WINTERP — The OSF/Motif Widget INTERPreter

Niels Mayer, Hewlett-Packard Labs, Palo Alto.

Winterp (pronounced Win-Terp) is a Widget Interpreter, an application development environment enabling rapid prototyping of graphical user-interfaces (GUI) through the interactive programmatic manipulation of user interface objects and their attached actions. *Winterp* is also an excellent platform for delivering extensible or customizable applications. By embedding a small, efficient Lisp interpreter with UI primitives within the delivered application, users and system integrators can tailor the static and dynamic layout of the UI, UI/application dialogue, and application functionality.

Winterp is a good tool for learning and experimenting with the capabilities of the *OSF/Motif* UI toolkit, allowing UI designers to more easily play "what if" games with different interface styles. *Winterp*'s implementation provides a compromise between the prototyping and extensibility advantages of Lisp environments, and the inefficiency and expenses of delivering Unix applications under environments such as Common Lisp. Typically, prototyping and customization are done entirely in interpreted Lisp; for delivery, efficiency-critical low-level code may be written in C and is easily exported to the interpreter as a new primitive.

The interpreter, based on David Betz's *XLISP*, provides a truly object oriented interface to the X11 toolkit Intrinsics (*Xt)* and the *OSF/Motif* widget set. *Winterp*'s use of a real programming language for customization/prototyping allows *Winterp*-based applications to be much more flexible than applications using lower-level and less-general languages provided by the X resource database, Brunecky&Smythe's Widget Creation Library (*WCL*), *OSF/Motif*'s *UIL* (user interface language), and Ousterhout's *TCL/Tk*.

To enable GUI-applications based on existing *Unix* facilities, *Winterp* provides primitives for collecting data from *Unix* processes, and facilities for interacting with other *Unix* processes. These facilities make it possible to glue together existing *Unix* functionality into a GUI based application with a relatively small amount of *Winterp*-Lisp.

An environment similar to *Winterp*'s already exists in the *Gnu-Emacs* text editor — *Winterp* was strongly influenced by *Gnu-Emacs*' successful design. In *Gnu-Emacs*, a mini-Lisp interpreter is used to extend the editor to provide text-browser style interfaces to a number of *Unix* applications (e.g. e-mail user agents, directory browsers, debuggers, etc.). Whereas *Emacs*-Lisp enables the creation of new applications by tying together C-implemented primitives operating on text-buffer UI objects, *Winterp*-Lisp ties together operations on graphical UI objects implemented by the *Motif* widgets. Both achieve a high degree of customizability that is common for systems implemented in Lisp, while still attaining the speed of execution and (relatively) small size associated with C-implemented applications.

Winterp features:

- Free with non-restrictive copyright. — available via anonymous *ftp* from *export.lcs.mit.edu*.
- Portable — entirely implemented via machine-independent C source and *X11/Xt/Motif* libraries.
- *XLISP* provides a simple *Smalltalk*-like object system.
- *OSF/Motif* widgets are real *XLISP* objects — widgets can be specialized via subclassing, methods added or altered, etc.
- Automatic storage management (via garbage collection) of *Motif/Xt/X* data.
- Contains facilities for simple "direct manipulation" of UI components.
- Interface to *Gnu Emacs*' lisp-mode allows code to be developed and tested without leaving the editor.
- Built-in RPC mechanism for inter-application communications, implemented via serverized, event-driven Lisp interpreter.

Features in upcoming version 1.14:

- Primitives interfacing to *XPM* library, providing a portable means of using color pixmaps in *Motif*. (user-contributed extension).
- High-level interface to *Xlib* drawing and text routines available as methods on *the Motif* drawing area widget. (user-contributed).
- Facility to allow multiple interactive *Unix* subprocesses to run without causing UI to block. (Similar to *Gnu-Emacs'* ability to run a subprocess via *process-filter/process-sentinel*).
- Interface to Don Libes' *expect* library further extends *Unix* subprocess capabilities.